
sphinx-notfound-page Documentation

Release 0.8

Manuel Kaufmann

Jul 13, 2022

CONTENTS

| | |
|-------------------------------------|-----------|
| 1 Why I need this extension? | 3 |
| Python Module Index | 13 |
| Index | 15 |

sphinx-notfound-page is a [Sphinx](#) extension to create custom 404 pages and help you to generate proper resource links (js, css, images, etc) to render the page properly.

This extension was originally developed to be used on [Read the Docs](#) but it can be used in other hosting services as well.

Online documentation: <https://sphinx-notfound-page.readthedocs.io/>

Source code repository (and issue tracker): <https://github.com/readthedocs/sphinx-notfound-page/>

Badges:

WHY I NEED THIS EXTENSION?

Sphinx does not create a 404 page by default. Although, you can create it by adding a simple `404.rst` file to your docs but...

If you are reading this documentation, you may already experienced the problem that all your images do no load, all your styles are broken and all the javascript events does not work, when accessing the 404 page of your documentation.

So, if you want to have a nice custom 404 page, you will probably want to use this extension to avoid this headache and let the extension handle these URLs properly for you.

1.1 Installation

Install the package

from PyPI

from GitHub

```
$ pip install sphinx-notfound-page
```

```
$ pip install git+https://github.com/readthedocs/sphinx-notfound-page
```

Once we have the package installed, we have to configure it on our Sphinx documentation. To do this, add this extension to your Sphinx's extensions in the `conf.py` file.

```
# conf.py
extensions = [
    # ... other extensions here
    'notfound.extension',
]
```

After installing the package and adding the extension in the `conf.py` file, you can build your documentation again and you will see a new file called `404.html` in your documentation's build output.

Warning: If you open the `404.html` file on the browser, you will see that all of the images and css does not display properly. This is because all the URLs are absolute and since the file is being rendered from `file://` in the browser, it does not know where to find those resources.

Do not worry too much about this, this is the expected behavior and those resources will appear once the docs are deployed.

If you can't see the `404.html` file using a local simple web server, it is most likely because they often don't support requests for 404 codes. Refer to the [Frequently Asked Questions](#) for more information.

1.2 Configuration

The default settings generate the most commonly-used URL pattern on [Read the Docs](#): if you have a resource at `_static/js/logic.js` and you generate a 404 page with the default settings, the URL for that resource will be `/en/latest/_static/js/logic.js`.

For other use cases, you can customize these configuration options in your `conf.py` file:

notfound_template

Template used to render the `404.html` generated by this extension.

Default: `'page.html'`

Type: string

notfound_context

Context passed to the template defined by `notfound_template`.

Default:

```
{
  'title': 'Page not found',
  'body': '<h1>Page not found</h1>\n\nThanks for trying.',
}
```

Type: dict

Note: If you prefer, you can create a file called `404.rst` and use reStructuredText to create the context of your `404.html` page. Add the `:orphan: metadata` to the top of `404.rst`, to silence the spurious document isn't included in any toctree warning.

notfound_pagename

Page name generated by the extension.

Default: `'404'`

Type: string

notfound_urls_prefix

Prefix added to all the URLs generated in the 404 page.

Default: `'/en/latest/'`

Type: string

Warning: Make sure this config starts and ends with a `/`. Otherwise, you may have unexpected behaviours.

Tip: The prefix can be completely removed by setting it to `None`.

notfound_default_language

Language used as default to generate all links to the resources.

Default: `READTHEDOCS_LANGUAGE` environment variable, if set, else `'en'`

Type: string

Note: All links generated will have this prefix (e.g. `/en/`). This setting works with `notfound_default_version` to create a prefix for all URLs.

Deprecated since version 0.5: `notfound_default_language` is deprecated. Use `notfound_urls_prefix` instead

notfound_default_version

Version used as default to generate all links to the resources.

Default: `READTHEDOCS_VERSION` environment variable, if set, else `'latest'`

Type: string

Note: All links generated will have this prefix (e.g. `/latest/`). This setting works with `notfound_default_language` to create a prefix for all URLs.

Deprecated since version 0.5: `notfound_default_version` is deprecated. Use `notfound_urls_prefix` instead

notfound_no_urls_prefix

URL prefixes skipped or included.

Default: `False`.

Type: bool

Note: If this option is set to `True`, the extension omits any prefix values from the URLs, including explicit values for `notfound_default_language` and `notfound_default_version`.

Deprecated since version 0.5: `notfound_no_urls_prefix` is deprecated. Use `notfound_urls_prefix` instead

1.3 How It Works

The extension subscribes to some events emitted by the Sphinx application. When these events are triggered, our functions are called and they manipulate the doctree and context passed to the template.

1.3.1 Events subscribed

There are 3 main events that this extension subscribes:

- `doctree-resolved`
- `html-collect-pages`
- `html-page-context`

Each one has an specific goal persuading the same objective: make all resources URLs absolutes.

doctree-resolved

After Sphinx has parsed our source files, this event is triggered. Here, we check if the page being rendered is `notfound_pagename` and in that case, we replace all the URLs for `.. image::`, `.. figure::` and other directives to point the right path.

html-collect-pages

After all HTML pages are collected and this event is emitted, we check for the existence of a 404 page already. If there is one, we do not need to do anything here. If the user has not defined this page, we render the template `notfound_template` with the context `notfound_context`.

html-page-context

Immediately before the template is rendered with the context, this event is emitted. At this point, we override `pathto`¹ function with our custom one that will generate the proper URLs. We also override `toctree`² key with the same content of the regular toctree but with all the URLs fixed to find the resources from the 404 page.

1.4 Get Involved

We appreciate a lot your interest on getting involved in this small project! Your help will benefit a lot of people around the world.

Please, if you want to collaborate with us, you can check out [the list of issues we have on GitHub](#) and comment there if you need further guidance or just send a Pull Request .

1.5 Who Is Using It?

These are some projects using this extension that you can take a look at to understand how they are configured and what's the behavior.

1.5.1 Read the Docs

- <https://github.com/readthedocs/readthedocs.org>
- Example URL
- Configuration file

¹ <https://www.sphinx-doc.org/page/templating.html#pathto>

² <https://www.sphinx-doc.org/page/templating.html#toctree>

1.5.2 PyVista

- <https://github.com/pyvista/pyvista>
- Example URL
- Configuration file

1.5.3 Write the Docs

- <https://github.com/writethedocs/www>
- Example URL
- Configuration file

1.5.4 The Carpentries

- <https://github.com/carpentries/handbook>
- Example URL
- Configuration file

1.5.5 attrs

- <https://www.attrs.org/>
- Example URL
- Configuration file

1.6 Frequently Asked Questions

1.6.1 Does this extension work with Read the Docs?

Yes.

Read the Docs should detect the `404.html` page generated by the extension automatically, and serve it when a user hits a not found page.

If you are using a Single Version project, you may want to set `notfound_urls_prefix` to `None`.

1.6.2 Does this extension work with GitHub pages?

Yes.

You may want to set `notfound_urls_prefix` to `None`, and then add `permalink: /404.html` in the `YAML` front matter.

If you are using the github provided domain, make sure to set the `notfound_urls_prefix` to your repository's name in between two forward slashes. For example if your repository is named `MyRepo`, then `notfound_urls_prefix = "/MyRepo/"`.

1.6.3 Does this extension work with Jupyter Book?

Yes.

You need to enable `sphinx-notfound-page` in your Jupyter Book `_config.yml` as a `custom extension`. It would look like similar to the following:

```
sphinx:
  extra_extensions:
    - notfound.extension
```

1.6.4 Why is my local web server not showing a 404.html?

Simple web servers, such as `http.server`, don't have a default handler for 404 codes, so it doesn't know to point to the generated `404.html`.

To see an example of adding a custom request handler for 404 codes, see: <https://stackoverflow.com/questions/22467908/python-simplehttpserver-404-page>

1.6.5 The answer I'm looking for is not here

Please, open an issue in our issue tracker, and let us know what's the problem you are having.

1.7 notfound

1.7.1 Submodules

`notfound.extension`

Module Contents

Classes

| | |
|--------------------------------|----------------------------------|
| <i>OrphanMetadataCollector</i> | Force the 404 page to be orphan. |
|--------------------------------|----------------------------------|

Functions

| | |
|---|--|
| <i>html_collect_pages</i> (app) | Create a <code>404.html</code> page. |
| <i>finalize_media</i> (app, pagename, templatename, context, doctree) | Point media files at our media server. |
| <i>doctree_resolved</i> (app, doctree, docname) | Generate and override URLs for <code>.. image:: Sphinx</code> directive. |
| <i>handle_deprecated_configs</i> (app, *args, **kwargs) | Handle deprecated configurations. |
| <i>validate_configs</i> (app, *args, **kwargs) | Validate configs. |
| <i>setup</i> (app) | |

exception `notfound.extension.BaseURLError`(*message: str, orig_exc: Exception = None, modname: str = None*)

Bases: `sphinx.errors.ExtensionError`

Exception for malformed base URI.

`notfound.extension.html_collect_pages`(*app*)

Create a `404.html` page.

Uses `notfound_template` as a template to be rendered with `notfound_context` for its context. The resulting file generated is `notfound_pagename.html`.

If the user already defined a page with pagename title `notfound_pagename`, we don't generate this page.

Parameters `app` (*sphinx.application.Sphinx*) – Sphinx Application

`notfound.extension.finalize_media`(*app, pagename, templatename, context, doctree*)

Point media files at our media server.

Generate absolute URLs for resources (js, images, css, etc) to point to the right URL. For example, if a URL in the page is `_static/js/custom.js` it will be replaced by `<notfound_urls_prefix>_static/js/custom.js`.

On the other hand, if `notfound_no_urls_prefix` is set, it will be replaced by `/_static/js/custom.js`.

Also, all the links from the sidebar (toctree) are replaced with their absolute version. For example, `../section/pagename.html` will be replaced by `/section/pagename.html`.

It handles a special case for Read the Docs and URLs starting with `/_/`. These URLs have a special meaning under Read the Docs and don't have to be changed. (e.g. `/_/static/javascript/readthedocs-doc-embed.js`)

Parameters

- `app` (*sphinx.application.Sphinx*) – Sphinx Application
- `pagename` (*str*) – name of the page being rendered
- `templatename` (*str*) – template used to render the page
- `context` (*dict*) – context used to render the page
- `doctree` (*docutils.nodes.document*) – doctree of the page being rendered

`notfound.extension.doctree_resolved`(*app, doctree, docname*)

Generate and override URLs for `.. image::` Sphinx directive.

When `.. image::` is used in the `404.rst` file, this function will override the URLs to point to the right place.

Parameters

- `app` (*sphinx.application.Sphinx*) – Sphinx Application
- `doctree` (*docutils.nodes.document*) – doctree representing the document
- `docname` (*str*) – name of the document

class `notfound.extension.OrphanMetadataCollector`

Bases: `sphinx.environment.collectors.EnvironmentCollector`

Force the 404 page to be orphan.

This way we remove the WARNING that Sphinx raises saying the page is not included in any toctree.

This collector has the same effect than `:orphan:` at the top of the page.

clear_doc(*self, app, env, docname*)

Remove specified data of a document.

This method is called on the removal of the document.

process_doc(*self, app, doctree*)

Process a document and gather specific data from it.

This method is called after the document is read.

merge_other(*self, app, env, docnames, other*)

Merge in specified data regarding docnames from a different *BuildEnvironment* object which coming from a subprocess in parallel builds.

`notfound.extension.handle_deprecated_configs`(*app, *args, **kwargs*)

Handle deprecated configurations.

Looks for old deprecated configurations, define the new ones and triggers warnings for old configs.

`notfound.extension.validate_configs`(*app, *args, **kwargs*)

Validate configs.

Shows a warning if one of the configs is not valid.

`notfound.extension.setup`(*app*)

`notfound.utils`

Module Contents

Functions

| | |
|---|---|
| <code>replace_uris</code> (<i>app, doctree, nodetype, nodeattr</i>) | Replace <i>nodetype</i> URIs from <i>doctree</i> to the proper one. |
|---|---|

`notfound.utils.replace_uris`(*app, doctree, nodetype, nodeattr*)

Replace *nodetype* URIs from *doctree* to the proper one.

If *nodetype* is an image (`docutils.nodes.image`), the URL is prefixed with `Builder.imagedir` and the original image path is added to `Builder.images` so it's copied using Sphinx's internals before finalizing the building.

Parameters

- **app** (*sphinx.application.Sphinx*) – Sphinx Application
- **doctree** (*docutils.nodes.document*) – doctree representing the document
- **nodetype** (*docutils.nodes.Node*) – type of node to replace URIs
- **nodeattr** (*str*) – node attribute to be replaced

1.7.2 Package Contents

`notfound.__version__ = 0.8.3`

PYTHON MODULE INDEX

n

`notfound`, 8
`notfound.extension`, 8
`notfound.utils`, 10

Symbols

`__version__` (in module `notfound`), 11

B

`BaseURIError`, 9

C

`clear_doc()` (*notfound.extension.OrphanMetadataCollector* method), 9

configuration value

- `notfound_context`, 4
- `notfound_default_language`, 4
- `notfound_default_version`, 5
- `notfound_no_urls_prefix`, 5
- `notfound_pagename`, 4
- `notfound_template`, 4
- `notfound_urls_prefix`, 4

D

`doctree_resolved()` (in module `notfound.extension`), 9

E

environment variable

- `READTHEDOCS_LANGUAGE`, 4
- `READTHEDOCS_VERSION`, 5

F

`finalize_media()` (in module `notfound.extension`), 9

H

`handle_deprecated_configs()` (in module `notfound.extension`), 10

`html_collect_pages()` (in module `notfound.extension`), 9

M

`merge_other()` (*notfound.extension.OrphanMetadataCollector* method), 10

module

- `notfound`, 8
- `notfound.extension`, 8

`notfound.utils`, 10

N

`notfound`

module, 8

`notfound.extension`

module, 8

`notfound.utils`

module, 10

`notfound_context`

configuration value, 4

`notfound_default_language`

configuration value, 4

`notfound_default_version`

configuration value, 5

`notfound_no_urls_prefix`

configuration value, 5

`notfound_pagename`

configuration value, 4

`notfound_template`

configuration value, 4

`notfound_urls_prefix`

configuration value, 4

O

`OrphanMetadataCollector` (class in `notfound.extension`), 9

P

`process_doc()` (*notfound.extension.OrphanMetadataCollector* method), 10

R

`READTHEDOCS_LANGUAGE`, 4

`READTHEDOCS_VERSION`, 5

`replace_uris()` (in module `notfound.utils`), 10

S

`setup()` (in module `notfound.extension`), 10

V

`validate_configs()` (*in module `notfound.extension`*),
[10](#)